代数的アルゴリズムに対する量子計算

AUTHOR(S):

武田, 邦敬; 甲斐, 博; 野田, 松太郎

CITATION:

武田, 邦敬 ...[et al]. 代数的アルゴリズムに対する量子計算. 数理解析研究所講究録 2003, 1335: 119-126

ISSUE DATE:

2003-07

URL:

http://hdl.handle.net/2433/43347

数理解析研究所講究録 1335 巻 2003 年 119-126

代数的アルゴリズムに対する量子計算

武田 邦敬*
KUNIHIRO TAKEDA
愛媛大学大学院 理工学研究科
GRADUATE SCHOOL OF SCIENCE AND ENGENEERING, EHIME UNIVERSITY

甲斐 博†
HIROSHI KAI
愛媛大学 工学部
DEPARTMENT OF COMPUTER SCIENCE, EHIME UNIVERSITY

野田 松太郎‡
MATU-TAROW NODA
愛媛大学 工学部
DEPERTMENT OF COMPUTER SCIENCE, EHIME UNIVERSITY

1 はじめに

新しい計算のパラダイムとして、量子コンピュータ [5] の概念が提唱され多くの関心が寄せられている。もし、量子コンピュータが実現すれば、現在の IT 社会の根幹となっている暗号技術に大きな課題を投げかける等々である。しかし、量子コンピュータの実現には多くの論理的・実験的課題があるため、この分野の研究は量子コンピュータの完成を想定してのアルゴリズム（量子アルゴリズム）に関するものが中心である。量子アルゴリズムの研究としては、Shor による整数の因数分解と離散対数問題に対する多項式時間アルゴリズム [10] と、未整理なデータベースに対する探索を目的とした Grover のアルゴリズム [6] が著名である。著者らは、Grover のアルゴリズムを発見的な多項式 GCD アルゴリズムである GCDHEU アルゴリズム [4] へ応用することを考察した [13]。Grover のアルゴリズムは解の存在率が低い場合に有効であるが、解の存在率が高いときには成功確率が下がってしまう欠点がある。したがって、GCDHEU アルゴリズムで扱うような解の存在率が高い問題に対しては、Grover のアルゴリズムをそのままの形で応用することは得策でないといえる。

河内らは、Grover のアルゴリズムを、解の存在確率が高い問題に対して有効に働くように改良した [12]。このアルゴリズムでは、通常の Grover のアルゴリズムで用いられるユニタリ変換を一般化することにより、誤り確率を極めて小さくすることに成功している。本研究では、[12] で提案された量子アルゴリズムを GCDHEU アルゴリズムへ応用することを考察する。

*kunihiro@hpc.cs.ehime-u.ac.jp
†kai@cs.ehime-u.ac.jp
‡noda@cs.ehime-u.ac.jp

2 量子計算の原理

量子計算の計算モデルとしては、量子チューリング機械 [5] と量子回路モデル [1, 11] が代表的である。両者の関係は [11] で詳しく議論されており、その等価性も証明されている。量子チューリング機械は計算モデルとして妥当であるが、具体的な計算過程とその状態を議論するためには量子回路モデルを用いる方が見通しがよい。そこで、この節では量子回路モデルを導入し、以降の議論をこのモデルの上で行うこととする。

量子回路は、1 量子ビットまたは複数量子ビットからなる系を抽象化した量子レジスタと、それに対する具体的な操作を記述するユニタリ変換 (量子ゲート) から構成される。量子ビットは、スピン 1/2 粒子系など系のオブザーバブルが 2 つの固有状態を持つ 2 状態系を数学的にモデル化したものである。量子ビットが持つ 2 つの固有状態を $|0\rangle, |1\rangle$ とおくと、観測前の量子ビットの状態は正規直交基底 $\{|0\rangle, |1\rangle\}$ によって張られる 2 次元複素 Hilbelt 空間の単位ベクトルとして記述される。即ち、量子ビットは

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \in \mathbf{C}^2 \tag{1}$$

なる重ね合わせ状態をとる。ただし、$|\alpha|^2 + |\beta|^2 = 1$ である。α, β は振幅と呼ばれ、量子ビットの状態を表すパラメータとなる。式 (1) の重ね合わせ状態にある量子ビットを観測すると、$|\alpha|^2$ の確率で $|0\rangle$、$|\beta|^2$ の確率で $|1\rangle$ という固有状態を得ることができる。観測は量子ビットの値を読み出すことに相当し、アルゴリズムの最後では必ず観測を行う必要がある。複数の量子ビットからなるレジスタの状態は、各量子ビットの状態を記述するベクトルのテンソル積で表される。即ち、n 量子ビットレジスタの状態は

$$|\psi_n\rangle = \bigotimes_{i=0}^{n-1} \Big[\alpha|0\rangle + \beta|1\rangle\Big] = \sum_{x\in\{0,1\}^n} \omega_x|x\rangle = \begin{bmatrix} \omega_{0\cdots0} \\ \vdots \\ \omega_{1\cdots1} \end{bmatrix} \in \mathbf{C}^{2^n} \tag{2}$$

となる。ただし、$\sum_{x\in\{0,1\}^n} |\omega_x|^2 = 1$ である。

各量子ビットに対する操作は、$\mathbf{C}^2$ 上のユニタリ変換によって表される。ユニタリ変換 U は、ノルムを変えない線形変換で、$UU^\dagger = U^\dagger U = I$ を満たす行列 U である。($U^\dagger$ は U の共役転置行列)。以下に示す Hadamard 変換 H は、1 量子ビットに対する重要なユニタリ変換である。

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad H : |0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad H : |1\rangle \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{3}$$

固有状態 $|\mathbf{0}\rangle \equiv |00\cdots0\rangle$ にある n 量子ビットレジスタの各ビットに Hadamard 変換 H を施すと、等しい振幅を持つ 2^n 個の重ね合わせ状態を作り出すことができる。

$$H^{\otimes n}|\mathbf{0}\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \tag{4}$$

n 量子ビットに対する任意のユニタリ変換は、1 量子ビットに対する任意のユニタリ変換と、以下に示す 2 量子ビットに対する制御 NOT ゲートと呼ばれるユニタリ変換 U_{CN} のみで構成できることが証明されている [1]。

$$U_{CN} \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad U_{CN} : |a\rangle|b\rangle \mapsto |a\rangle|b \oplus a\rangle \tag{5}$$

式 (5) における $\oplus$ は mod 2 上の加算であり、古典計算の排他的論理和 XOR と等価である。また、ある関数 $f(x):\{0,1\}^n \to \{0,1\}^m$ に対し、次のようなユニタリ変換 U_f を考えることができる。

$$U_f : |a\rangle|b\rangle \mapsto |a\rangle|b \oplus f(a)\rangle \tag{6}$$

ただし、$a \in \{0,1\}^n$, $b \in \{0,1\}^m$, $\oplus$ は $\{0,1\}^m$ 上の加算である。特に $b = \mathbf{0}$ とすると $U : |a\rangle|\mathbf{0}\rangle \mapsto |a\rangle|f(a)\rangle$ となり、$f(a)$ の値を計算する変換になる。

量子アルゴリズムの計算量に関しては様々な議論があるが、回路上の基本ゲートの数を考えるのが一般的である。しかし、量子コンピュータを実現する具体的なデバイスイメージが定まっていない現段階では、代数的アルゴリズムで用いられるすべての演算について、その量子回路を考えることは極めて困難である。そこで、量子チューリング機械と通常のチューリング機械が計算可能性において等価であることと、量子回路モデルが古典計算に対してユニバーサルであるという事実に基づき、代数的アルゴリズムで使用する古典的な演算は、量子回路モデル上でも同様に実行できると仮定して以下の議論を行う。

3 Groverのアルゴリズムと一般化手法

3.1 Groverのアルゴリズム

Groverのアルゴリズムは、N 個のデータからなる未整理のデータベースから、求める一つのデータを探し出すアルゴリズムである。Groverのアルゴリズムではこの問題を次のように置き換える。$N = 2^n$ とし、N 個のデータを $\{0,1\}^n$ でラベル付けし、次のような関数 $f:\{0,1\}^n \to \{0,1\}$ を考える。

$$f(x) = \begin{cases} 1 & x\text{が解である} \\ 0 & x\text{が解でない} \end{cases} \tag{7}$$

データベース探索問題は $f(x_0) = 1$ を満たす唯一の $x_0 \in \{0,1\}^n$ を見つけることに帰着する。Groverのアルゴリズムは、この探索問題を解くために $O(\sqrt{N})$ 回の f の評価を要する。一方、古典計算では f をランダムに参照していくしかないため、$\Theta(N)$ 回の f 評価が必要になる。したがって、古典計算に比べて平方根分速く解を探索することができる。

Groverのアルゴリズムでは、関数 f の全ての入力の重ね合わせ状態 $H|\mathbf{0}\rangle = \sum_{x\in\{0,1\}^n} |x\rangle$ に対し、以下で定義するユニタリ変換 G を $O(\sqrt{N})$ 回施すことにより高確率で $f(x_0) = 1$ をみたす解 x_0 を観測できる状態を作る。

$$G \equiv -H^{\otimes n} I_0 H^{\otimes n} I_f, \quad I_f : |x\rangle \mapsto (-1)^{f(x)}|x\rangle, \quad I_0 : |\mathbf{0}\rangle \mapsto -|\mathbf{0}\rangle \tag{8}$$

ユニタリ変換 G を Grover 反復という。G を 1 回適用するごとに 1 回の関数 f の評価が行われる。$H|\mathbf{0}\rangle$ に G を $O(\sqrt{N})$ 回施した量子レジスタを観測したとき、$f(x_0) = 1$ を満たす x_0 を得る確率は $\Omega(1 - 1/N)$ であることが保証されている [7]。Groverのアルゴリズムを次に示す。

アルゴリズム 1 (Groverのアルゴリズム)
入力： 関数 $f:\{0,1\}^n \mapsto \{0,1\}$
出力： $f(x_0) = 1$ を満たす x_0

procedure Grover

1. 基底状態にある n 量子ビットレジスタに対して Hadamard 変換を施し、関数 f のすべての入力の重ね合わせ状態を作る。
2. Grover 反復 G を $O(\sqrt{N})$ 回適用する。

3. 量子レジスタ観測して解を得る。

end;

Grover のアルゴリズムは、解の個数が複数存在する場合にも適用できるように拡張されている [2, 7]。解の個数が t であるとき、G を $O(\sqrt{N/t})$ 回施すことによって高確率で解を得ることができる。また、Grover 反復の適用回数は解の存在率に依存するが、解の存在率が未知の場合においても同様の計算量で解を求めることができるアルゴリズムが提案されている [2]。

Grover のアルゴリズムで注意すべき点は、解の存在率が低い問題に対して有効に働くように設計されていることである。解の存在率が高くなると古典的なランダムサンプリングに近い動作をとるようになり、成功確率が下がってしまうことも知られている。

3.2 量子サンプリング

Grover のアルゴリズムを、古典的なランダムサンプリングの成功確率を増加させる手法に応用することができる。これを、振幅増幅または量子サンプリングと呼ぶ [3]。[3] では、Grover 反復 $G = -H^{\otimes n} I_0 H^{\otimes n} I_f$ を一般化する手法をとっている。一方、回転変換である I_0 と I_f のみを一般化し、Grover のアルゴリズムの誤り確率を小さくするように改良されたアルゴリズムが提案されている [12]。このアルゴリズムでは、ユニタリ変換の反復を行わず、関数 f の評価を 1 度だけ行っていることが特徴である。[12] では、Grover 反復 $G = -H^{\otimes n} I_0 H^{\otimes n} I_0$ を、次のように一般化している。

$$Q \equiv -H^{\otimes n} S_0 H^{\otimes n} S_f, \quad S_f : |x\rangle \mapsto e^{i\theta \cdot f(x)} |x\rangle, \quad S_0 : |\mathbf{0}\rangle \mapsto e^{i\theta} |\mathbf{0}\rangle \tag{9}$$

Grover のアルゴリズムの初期状態 $H|\mathbf{0}\rangle$ に対し、ユニタリ変換 Q を 1 回だけ適用して観測を行ったとき、$f(x) \neq 1$ なる x、即ち誤った解を観測する確率 P_ϵ は次のように計算できる [12]。

$$P_\epsilon = \{1 - 2\cos\theta + 2(1 - \cos\theta)\epsilon\}^2 \epsilon, \quad \epsilon = \frac{N - t}{2^n} \tag{10}$$

ここで、ϵ は古典的なランダムサンプリングを行ったときの誤り確率である。式 (10) より、$t \geq 1/4$ である必要がある。$\theta = \pi/3$ と固定すると、誤った解を観測する確率は ϵ^3 となり、古典的なランダムサンプリングを行ったときの ϵ に比べて極めて小さくすることができる。ゆえに、このアルゴリズムを確率的アルゴリズムに応用することで、アルゴリズムが含む誤りを圧縮することが期待できる。

4 GCDHEU アルゴリズム

GCDHEU アルゴリズムでは、整数係数多項式の GCD を求める発見的なアルゴリズムである [4]。このアルゴリズムでは、一つの大きな整数を評価点として整数上で GCD を計算し、得られた整数から多項式を再構築して GCD を求める。入力が 1 変数多項式の場合のアルゴリズムの概略を以下に示す。

アルゴリズム 2
入力: $\mathbf{Z}$ に関して原始的な 1 変数多項式 $a, b \in \mathbf{Z}[x]$
出力: $\gcd(a, b)$ または fail_flag

procedure GCDHEU(a, b)
  $\xi \leftarrow 2 \times \min\left(\|a\|_\infty, \|b\|_\infty \right) + 2$;
  for i **from** 1 **to** *Limit*1 **do**

```
    if length(ξ) × max( deg_x(a), deg_x(b) ) > Limit2 then
      RETURN_TO_TOP_LEVEL(fail_flag) fi;
    g ← genpoly(igcd( φ_{x-ξ}(a), φ_{x-ξ}(b) ), ξ, x);
    if pp(g) | a and pp(g) | b then return g fi
    ξ ← selsect(ξ)
  od;
  return fail_flag
end;
```

GCDHEUアルゴリズムの特徴は、評価点の取り替えに次のような発見的手法を導入している点である。

$$\xi^{(1)} \leftarrow 2 \times \min\left(\|a\|_\infty, \|b\|_\infty \right) + 2; \quad \xi^{(i+1)} \leftarrow \mathrm{quo}(\xi^{(i)} \times 73794, 27011); \quad (i \geq 1) \tag{11}$$

評価点の選択をランダムに行うと、アルゴリズム2は確率的アルゴリズムであると考えることができる。GCDHEUアルゴリズムは次のような性質を持つことが解っている [13]。

- 評価点の取り替え、すなわち探索回数は数回程度で完了する。しかし、数回の探索で評価点が増大するため急激な速度低下が起こっている。
- 計算に成功する評価点の存在率は高い。
- 評価点の選択を1ずつ増加させる方法をとると、評価点の増大を防ぐことができる。ただし、この場合は探索回数が増える可能性がある。

5 量子アルゴリズムの応用

著者らは、Groverのアルゴリズムを GCDHEUアルゴリズムへ応用することを考察した [13]。Groverのアルゴリズムは $\{0,1\}$ を返す関数に対する探索問題を解くアルゴリズムである。そこで、次のような関数 $F : \mathbf{Z} \to \{0,1\}$ に対する探索を行って成功する評価点を探索し、その評価点を使って GCD を計算する必要がある。

$$F(x) = \begin{cases} 1 & g = \mathrm{genpoly}\left(\mathrm{igcd}(\phi_{v-(x+\xi^{(1)})}(a), \phi_{v-(x+\xi^{(1)})}(b)), x + \xi^{(1)}, v\right) \\ & \text{なる } g \text{ に対し, } g | \mathrm{pp}(a) \text{ かつ } g | \mathrm{pp}(b) \\ 0 & \text{それ以外} \end{cases} \tag{12}$$

しかしながら、この応用には次のような問題点がある。

- 関数 F の計算が Grover 反復に含まれており繰り返し実行される。そのため、古典的な GCDHEU アルゴリズムに対する優位性がない。
- 計算に成功するような評価点の存在率が比較的高いため、Groverのアルゴリズムの応用として問題不適当である。

したがって、Groverのアルゴリズムの応用として GCDHEUアルゴリズムを考えるのは、問題として適当でないといえる。そこで、本研究では、GCDHEUアルゴリズムをある種の確率的アルゴリズムとして考えて量子サンプリングを導入し、アルゴリズムの過程で生じる誤りを圧縮することを期待した量子的な GCDHEUアルゴリズムを提案する。

量子サンプリングの枠組みを用いた、成功する評価点を探索する量子的なサブルーチンを用意する。これを次に示す。

アルゴリズム 3
入力： $\mathbf{Z}$ に関して原始的な 1 変数多項式 $a, b \in \mathbf{Z}[x]$, 探索範囲 (量子ビット数) m
出力： $F(x) = 1$ を満たす x, 成功確率 $1 - \epsilon^3$ （ϵ は古典的なランダムサンプリングを行ったときの誤り確率）
procedure QuantumSubroutine(a, b, m)

1. 基底状態にある m 量子ビットレジスタに Hadamard 変換を適用し、重ね合わせ状態を作る。
2. ユニタリ変換 $Q_F \equiv -H^{\otimes m} S_0 H^{\otimes m} S_F$ を一回だけ適用する。
3. 量子レジスタを観測して出力 $\xi_i \in \mathbf{Z}$ を得る。

end;

このサブルーチンを利用した量子的な GCDHEU アルゴリズムを以下に示す。

アルゴリズム 4
入力： $\mathbf{Z}$ に関して原始的な 1 変数多項式 $a, b \in \mathbf{Z}[x]$
出力： $\gcd(a, b)$ または fail_flag
procedure QuantumGCDHEU(a, b)
  $\xi^{(1)} \leftarrow 2 \times \min\left(\|a\|_\infty, \|b\|_\infty \right) + 2$;
  $m \leftarrow \min(\lceil \log(\mathrm{quo}(\xi^{(1)} \times 73794, 27011) - \xi^{(1)}) \rceil, 10)$;
  for i **from** 1 **to** *Limit*1 **do**
    if $m >$ *Limit*2 **then**
      RETURN_TO_TOP_LEVEL(fail_flag) **fi**;
    $\xi \leftarrow$ QuantumSubroutine$(a, b, m) + \xi^{(1)}$;
    $g \leftarrow \mathrm{genpoly}\left(\mathrm{igcd}\left(\phi_{x-\xi}(a), \phi_{x-\xi}(b)\right), \xi, x\right)$;
    if $\mathrm{pp}(g) \mid a$ and $\mathrm{pp}(g) \mid b$ **then return** g **fi**;
    $m \leftarrow m + 1$
  od;
  return fail_flag
end;

アルゴリズム 4 では、まず初めに探索範囲を定め、量子的なサブルーチン (アルゴリズム 3) によって計算に成功する評価点を高確率で探索する。計算に失敗した場合は、量子ビットを 1 ビット増やして、即ち、探索空間を 2 倍に広げて探索からやり直す。評価点の探索範囲を 2 倍にすると失敗する確率が約 1/2 減少することが知られているため [4]、このような方法をとっている。提案したアルゴリズムは、一部の計算手続きを 2 度行う必要があるという問題があるものの、古典的な GCDHEU アルゴリズムに比べ 1 回の探索における誤りを極めて小さく抑えることができる点で有利である。本研究では、数式処理システム Risa/Asir 上で量子的な GCDHEU アルゴリズム (アルゴリズム 4) に対するシミュレータを作成し、次の例題を用いて動作の考察を行った。

例 1 $g_1 = \gcd(a_1, b_1)$

$$
\begin{aligned}
a_1 &= 8x^{29} + 22x^{28} + 36x^{26} + 171x^{25} + 206x^{24} + 144x^{23} + 280x^{22} + 197x^{21} + 309x^{20} + 499x^{19} \\
&\quad + 320x^{18} + 763x^{17} + 917x^{16} + 312x^{15} + 800x^{14} + 1143x^{13} + 516x^{12} + 707x^{11} + 997x^{10} \\
&\quad + 754x^9 + 483x^8 + 322x^7 + 776x^6 + 144x^5 + 320x^4 + 144x^3 + 320x^2 \\
b_1 &= 8x^{28} + 22x^{27} + 4x^{23} + 53x^{22} + 8x^{20} + 42x^{19} + 21x^{17} + 18x^{16} + 44x^{15} + 21x^{14} + 9x^{11} + 20x^{10} \\
g_1 &= 4x^{15} + 11x^{14} + 21x^9 + 4x^7 + 21x^6 + 9x^3 + 20x^2
\end{aligned}
$$

例 2 $g_2 = \gcd(a_2, b_2)$

$$
\begin{aligned}
a_2 &= 7x^{28} + 34x^{27} + 54x^{26} + 62x^{25} + 77x^{24} + 18x^{23} + 2x^{22} + 89x^{21} + 306x^{20} + 352x^{19} + 606x^{18} \\
&\quad +648x^{17} + 280x^{16} + 490x^{15} + 268x^{14} + 427x^{13} + 498x^{12} + 722x^{11} + 661x^{10} + 553x^9 \\
&\quad +448x^8 + 138x^7 + 528x^6 + 178x^5 + 198x^4 + 152x^3 + 275x^2 + 210x \\
b_2 &= 6x^{29} + 12x^{28} + 6x^{25} + 23x^{24} + 22x^{23} + 66x^{22} + 48x^{21} + 72x^{19} + 74x^{18} + 185x^{17} + 88x^{16} \\
&\quad +120x^{15} + 174x^{14} + 136x^{11} + 194x^{10} + 77x^9 + 96x^8 + 64x^4 + 56x^3 \\
g_2 &= x^{16} + 2x^{15} + 11x^9 + 8x^8 + 12x^6 + 8x^2 + 7x
\end{aligned}
$$

例 1 と例 2 を計算したときの動作の比較を、それぞれ表 1 と表 2 に示す。例 1 は誤りが比較的多い例である。量子サンプリングを用いることによって誤り確率が小さくなり失敗なく計算ができている。一方、例 2 は成功する評価点の存在率が高いが、古典的な GCDHEU アルゴリズムでは失敗が続いて評価点が大きくなってしまう例である。このような例に対して量子サンプリングを適用すると誤り確率をほとんど 0 に抑えることができ、失敗なく計算することができる。

表 1: 例 1 による動作の比較

	GCDHEU	Quantum GCDHEU
評価点の取り替え	2 回	なし
GCD の計算回数	3 回	2 回
計算に成功した評価点	(108 → 295 →) 805	119
誤り確率	0.56250	0.17797

表 2: 例 2 による動作の比較

	GCDHEU	Quantum GCDHEU
評価点の取り替え	2 回	なし
GCD の計算回数	3 回	2 回
計算に成功した評価点	(390 → 1065 →) 2909	485
誤り確率	0.12500	0.0019531

6 まとめ

本研究では、量子計算の代数的アルゴリズムへの応用を考え、新しい量子アルゴリズムを提案した。提案した量子的な GCDHEU アルゴリズムは、計算手続きの一部を 2 度行う必要がある問題があるが、古典的な GCDHEU アルゴリズムに比べて誤り確率をかなり小さく抑えることができる点で優れている。

現在、量子アルゴリズムが効果的に働く問題群を代数的にモデル化する研究が盛んに行われている。HSP(Hidden Subgroup Problem)[8, 9] はその代表である。量子計算の有効性が十分解っていない現段階では、このような試みは量子計算の可能性を探る上で非常に重要である。そこで、代数的アルゴリズムで扱う問題を代数的にモデル化し、代数的構造の観点から量子計算への応用を考えることは、両者の接点を見つ

ける鍵になると考えられる。例えば、HSP の理論を多項式環上のイデアルを扱えるように発展させることができれば、グレブナ基底の計算などへの応用も考えられるようになり、量子計算と代数的アルゴリズムとの接点がより明確な形で議論できるようになるであろう。

参 考 文 献

[1] A.Barenco, C.H.Bennett, R.Clive, D.P.Di-Vincenzo, N.Margolus, P.W.Shor, T.Sleator, J.A.Smolin and H.Weinfurter : Elementary gates for quantum computation, *Physical Review*, A, vol.52, no.5, pp.3457–3467, 1995

[2] M.Bøyer, G.Brassard, P.Høyer and A.Tapp : Tight bounds on quantum searching, *Proceedings of PhysComp'96*, 1996

[3] G.Brassard, P.Høyer, M.Mosca and A.Tapp : Quantum amplitude amplification and estimation, `http://xxx.lanl.gov/archive/quant-ph/0005055`, 2000

[4] B.W.Char, K.O.Geddes and G.H.Gonnet : GCDHEU: Heuristic Polynomial GCD Algorithm Based On Integer GCD Computation, *Journal of Symbolic Computation*, vol.7, pp.31-48, 1989

[5] D.Deutsch : Quantum Theory, the Church-Turing Principle, and the Universal Quantum Computer, *Proceedings of Royal Society London*, Vol.A400, pp.97–117, 1985

[6] L.K.Grover : A fast quantum mechanical for database search, *Proc. Annual ACM Symposium on Theory of Computing*, pp.212-219, 1996

[7] L.K.Grover : A framework for fast quantum mechanical algorithms, *Proceedings of 30th ACM Symposium on Theory of Computing*, pp.53–63, 1998

[8] R.Jozsa : Quantum algorithms and the Fourier transform, *Proceedings of Royal Society London A*, pp.323–337, 1998

[9] M.Mosca and A.Ekert: The Hidden subgroup problem and eigenvalue estimation on a quantum computer, *Lecture Notes in Computer Science*, vol.1509, pp.174–188, 1999

[10] P.W.Shor : Algorithms for Quantum Computation: Discrete Logarithms and Factoring, *Proc. 35th Annual Symposium on Foundations of Computer Science*, pp.124–134, 1994

[11] A.Yao : Quantum Circuit Complexity, *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society Press, pp.352–360, 1993

[12] 河内 亮周, 山下 茂, 岩間 一雄 : 占有問題に対する量子アルゴリズム, 信学技報, COMP, 2002-56, pp.31-36, Nov. 2001

[13] 武田 邦敬, 甲斐 博, 野田 松太郎 : 量子アルゴリズムを用いた多項式 GCD の計算, 数理解析研究所講究録「Computer Algebra – Algorithms, Implementations and Applications」掲載予定, 2003